PCT

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(54) Title: A METHOD FOR MOBILE TEXT ENTRY
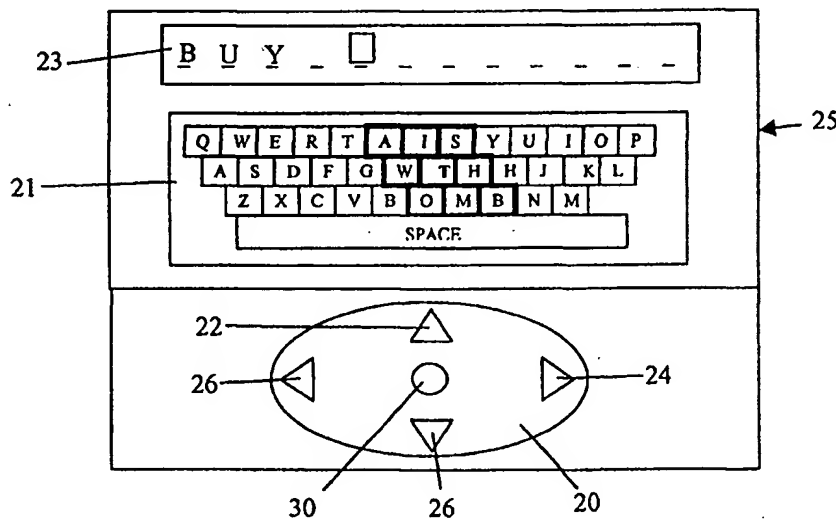
(57) Abstract

    A method for inputting characters in a device having a character display and a navigation key responsive to user manipulation for navigating a cursor in the character display. The characters are displayed in a predetermined layout on the display for character input. The layout contains a fixed set of characters as well as a dynamic subset of characters. The dynamic subset is determined by a previously entered single character or string of characters. A character is selected by a user from one of the aforementioned sets. It is selected by navigating a cursor to the desired character. In a preferred embodiment, the device may include a pager, a cellular telephone, or the like.

A Method for Mobile Text Entry

This invention relates to hand-held personal electronic devices such as cellular telephone systems, pagers, and the like, and in particular it relates to text entry for such devices.

5

BACKGROUND OF THE INVENTION

There is an increasingly strong demand around the world for portable communication devices such as cellular phone and pagers. As technology advances these products are becoming more portable and more affordable. Such devices have been able to receive text

10 messages for some time now, but with the growing ubiquity of email, products that have the capability to send text messages are now beginning to surface.

The requirement of mobility imposes a unique set of restrictions on the design of a text entry strategy. Typically there is only enough space on the device for a very few extra buttons. Therefore, a full keyboard, even a miniature one, is not feasible. Also, a user may

15 not always have both hands free when operating the device, so it is desirable that text can be input with only one hand. In this disclosure, the term "mobile text entry" refers to text entry limited by this set of restrictions.

One method that addresses the issue of mobile text entry is a telephone keypad method. It is well known that keys 2 through 9 on a telephone keypad contain three reference

20 letters of the alphabet. Entering a letter involves selecting the key it appears on as well as specifying which of the three letters is desired. The letter selection can be achieved in several different ways.

A second method for text entry is referred to as the date stamp method. The method is so named because, as with a date stamp, a desired character is made visible by rotating a

25 wheel containing an entire set of characters. Cursor keys are used to navigate a cursor in a display while increment and decrement keys are used to cycle sequentially through the character set at the cursor position. Players of video arcade games are familiar with this technique, which is used to input the player's initials into a list of high scorers. Selection of a character is performed implicitly by simply moving the cursor to another location. In effect,

30 the cursor position is a moveable editing window of one character.

A third method is referred to as a soft keyboard method. In this method a display is functionally split into an output and an input section. The input section of the display either

shows the full character set or allows for parts of the character set to be viewed by scrolling. The character output appears in a ticker tape fashion along one or more lines in the display section. Arrow keys are used to move a cursor around the character set and characters are selected by explicitly depressing a select key.

5   For example, an on-screen keyboard, sometimes called a soft keyboard, is a character set that appears on a device's Liquid Crystal Display (LCD) or on a computer monitor. With an on-screen keyboard, the advantage of have a fixed layout is familiarity. As familiarity increases, the time to locate a character in the layout approaches zero. The disadvantage of the fixed layout is that the distance between characters, and thus the time to move the cursor,

10   stylus or finger from one character to the next, is often high.

The problem with the aforementioned techniques is that an excessive number of keystrokes are required for text entry. The fewer the keystrokes required, the more efficient the text entry system. By bringing likely next characters closer to the cursor, keystrokes are reduced, which can lead to an increase in entry speeds. This can be accomplished by

15   surrounding the cursor with a dynamic region. The larger the number of dynamic positions that are available on the keyboard, the more likely a desired character will occur within the available positions.

In U.S. patent number 5,128,672 a dynamic keyboard is disclosed. However, the keyboard disclosed requires the presence of numerous physical keys, which is not practical

20   for use with cellular phones and pagers. In U.S. patent 5,797,098 a dynamic keyboard is also disclosed, but this keyboard only displays several options at a time. If the desired character is not presented on the display, it is necessary to scroll to the following screen and repeat the search for the character. This may need to be repeated several times until the character is found which can be tedious and frustrating for the user.

25   It is an object of the present invention to obviate or mitigate some of these problems.

SUMMARY OF THE INVENTION

In general terms, the present invention provides a method for inputting characters in a device having a character display, and a navigation key responsive to user manipulation for navigating a cursor in the character display. The method comprises the steps of:

30   (a) displaying a character set having a predetermined layout on the display, the layout being fixed for one or more character input;

        (b) displaying a subset of the character set in a dynamic layout wherein the subset of characters are determined by at least one preceding input character; and

(c) selecting at least one of the characters from the layouts by navigating the cursor to a desired character.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other embodiments of the invention will now be described by way of example only, with reference to the accompanying drawings in which:

Figure 1 is a schematic top view of a hybrid keyboard layout according to an embodiment of the invention;

Figure 2 is a schematic diagram showing a pager having a cursor control key;

Figure 3 is a top view of part of a keyboard according to the present invention;

Figure 4 is an example of a character probability look-up table;

Figures 5a and 5b are examples of a probability matrix;

Figures 6a and 6b are schematic diagrams of a layout after different keys are selected;

Figure 7 is an illustration of characters assigned to particular positions in the dynamic region;

Figures 8a and 8b are schematic diagrams of a layout incorporating the fixed positions of figure 7 after different keys are selected;

Figures 9a, 9b, 9c, 9d, 9e, and 9f illustrate various positioning patterns for a dynamic region;

Figures 10a, 10b, and 10c are schematic diagrams representing various dynamic region layouts;

Figures 11a, 11b, 11c, 11d and 11e illustrate various positions and arrangements of the dynamic region layout within the fixed layout;

Figure 12 illustrates a positioning algorithm for the dynamic region;

Figures 13a and 13b are flow charts showing the display of characters in the dynamic region;

Figures 14a and 14b show steps for generating a list for display in the dynamic region;

Figure 15 is a flow diagram showing the steps for arranging characters in the dynamic region.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

For convenience in the following description, like numerals refer to like structures in the drawings.

5   Referring to figure 1 a top view of a hybrid keyboard layout according to an embodiment of the invention is shown generally by numeral 10. The keyboard is comprised of a static key region 12 and a dynamic key region 14. In the keyboard layout of figure 1, the static region is comprised of a standard QWERTY keyboard layout. The dynamic region is defined by nine keys or a 3X3 matrix, which is inserted between the (T, G, B) and (Y, H, N) columns of

10   the QWERTY keyboard.

The hybrid keyboard 10 is an extension of an on-screen keyboard method of text entry that combines the benefits of both familiarity and reduced keystrokes. Text entry is performed by either moving a cursor around the display and selecting the cursor-highlighted character or, if implemented on a touch-sensitive display, by selecting characters directly by

15   applying pressure with either a stylus or finger to the touch-sensitive display.

It is possible to support both of the above methods in a single device. In the case of a cursor implementation, cursor movement may be supported by many different devices. Some of the more popular of these devices are:

(a) a set of arrow keys, typically containing four arrow keys and a selection key;

20   (b) a single pivoting button found on many pagers;

(c) a joystick;

(d) a touchpad; or

(e) a trackpointer (the device on IBM Thinkpads).

Figure 2 is a schematic diagram of a pager type device having a multi-axis pivoting

25   button 20 which may be used to control a cursor. The pager has a display area 25, which is divided into a message display area 23 and a character input area 21. The character input area incorporates a soft keyboard according to the embodiment shown in figure 1. When the button 20 is pressed on the top 22, right 24, bottom 26 or left 28, it moves the on-screen cursor 16 in that direction. When the button 20 is pressed in the centre 30 the character at the

30   cursor position 16 is selected. When a character is selected, it appears in display area 21, which is referred to as the output section.

At startup, the hybrid keyboard, shown in figure 3, is normally initialised with the cursor at a home position 16. A user looks for the desired character in the dynamic region 14.

If the character is found the user can navigate to it using the cursor control and then select it. Otherwise, if the character is not found, the user can simply and quickly select it from its known position in the fixed part of the layout 12.

After each character is entered, the cursor snaps back to the home position. This saves the user keystrokes by returning the cursor to the centre of the dynamic region 14 of the keyboard. It also eliminates the computation associated with repositioning the dynamic keys 14 around a new cursor position.

Various techniques may be implemented to further increase entry speed. For example, simplifying the use of the space character has the potential to positively effect entry speeds due to its high probability of occurrence (18% in English). In the case of cursor navigation, the space character may be implicitly selected by simply moving the cursor to a "virtual" space bar 18 located outside the layout of characters. Thus, the placement of the cursor 16 over the space bar 18 selects the space character without have to press the select button 30, after which the cursor 16 automatically snaps back to the home position 15.

The delete key is another common occurrence in text entry, and may be treated in a similar fashion to the space character, as may other control functions.

Referring once again to figure 1, the entire on-screen keyboard thus consists of standard alphanumeric characters in a fixed QWERTY layout 12, and a dynamic region 14 which display a given set of characters that is dependent on the key pressed.

For example, assume that a character γ is entered (either on the fixed or dynamic region of the keyboard). The system fills the layout positions in the dynamic region 14 with the first n characters in a list L, where n is the size of the dynamic region and L is a prioritized list of characters. The characters in the list L are organized such that the most likely character to follow the character γ appears first in the list L whereas the least likely character to follow the character γ appears last in the list.

The list L may be constructed in a number of ways. In one embodiment, the system is provided with a lookup table or database wherein each character is associated with a sorted list of the character set (including γ, as can be the case when a double letter is entered), as shown in figure 4. The appropriate list is read into the memory corresponding to γ.

In a second embodiment, the system may build L at runtime by consulting a probability (or frequency) matrix P that stores the likelihood of any character to follow a sequence of $m - 1$ characters, the last of which is γ. The value m is the statistical order which

measures the average predictability of the next letter, given that we know the preceding (m –

1) letters. That is, (m – 1) is the number of previously entered characters for which

Memory probability data is stored. A second-order (m = 2) embodiment contains lists or a matrix

based on 2-character sequences (digram) probabilities of the language in question. A third-

5   order (m = 3) embodiment contains lists or a matrix based on 3-character sequences (trigram)

probabilities of the language in question and so on.

The higher that the value of m is, the higher the number of rows in P will be. P is an

s^{m-1} by s matrix, where s is the size of the character set. The row headings of matrix P

contain all the possible permutations of (m – 1) elements of the character set. The column

10   headings of matrix P each contain an element of the character set. An example of how the

matrix is set up for an m = 3 embodiment is shown in figure 5a.

The number in each cell of the matrix contains the likelihood of the character in that

cell's column heading to follow the sequence of characters in its row heading. Based on the

user's input, a row is selected and its columns are sorted by the values in the cells. The

15   columns are arranged from most likely to occur to least likely to occur, and the first n

characters are then included in the dynamic region 14 of the keyboard. In the simplest case,

m = 2, P is an s by s matrix. An example of a matrix with m = 2 is illustrated in figure 5b.

As m increases, the number of rows in P increases exponentially.

In this embodiment, an English language character set is implemented and therefore

20   the fixed layout 12 is chosen to be the best known English configuration, which happens to

be the QWERTY layout. As for the dynamic region 14, there is a point at which the difficulty

of locating a character in a new arrangement of the dynamic region 14 outweighs the benefit

of reduced distance to the next character. For m = 2, for example, this point occurs at about 9

dynamic positions. In English text entry, the desired character occurs in a dynamic part of

25   size 9 just less than 90% of the time. There is thus little if any benefit to be derived by the

addition of more dynamic positions. Therefore n = 9, and the nine most likely characters to

follow γ are placed in the dynamic region of the keyboard.

Initially the dynamic region 14 may be set up as if the last character entered was a

space. Therefore the nine characters displayed to the user are the nine most likely to begin a

30   new word. The most likely character is placed at the home position 15 of the dynamic key

set with the remaining eight characters placed around it. The rest of the characters are filled

in as shown in figure 12.

The positioning of the outer eight characters is also influenced by which characters are placed there. There are certain letters that will occur frequently in the dynamic keys, particularly the vowels. However, although the same letters may reoccur they may not necessarily have the same probability of reoccurring. Therefore, based on the previously

5   mentioned placement scheme the same letters will reoccur in different locations. This is illustrated by figures 6(a) and 6(b). Figure 6(a) illustrates the arrangement of the dynamic region 14 after the letter 'G' is selected and figure 6(b) illustrates the arrangement if the letter 'S' is selected. It is clear that many of the same letters are used, but they are placed at different places within the dynamic region 14. This can be improved by providing some

10   characters with assigned positions regardless of their probability of occurrence, as long as they are found between positions two and nine, inclusively, in L. The most probable next character is always placed at the cursor's home position 15.

Therefore it is necessary to choose the eight characters that most commonly occur in the dynamic region 14 and assign them fixed positions within the eight key slots surrounding

15   the home position 15. While this slightly violates the rule of positioning characters according to their position in L, it reduces the user's uncertainty as to where a letter is going to appear. The number eight was chosen to avoid a situation of having to decide which of two or more characters should occupy a particular slot. For illustrative purposes only, and assuming that the eight most common characters are:

20   A, E, I, O, U, S, N, L

then their fixed positions are shown in figure 7. Should one of these letters occur in the dynamic region 14, then it will always occupy the same spot unless it is the most probable next character, which always occupies the home position 15. Figures 8(a) and 8(b) illustrate how fixed positions would affect figures 6(a) and 6(b), respectively. Although figure 6(a)

25   contains the same letters as figure 8(a), and figure 6(b) contains the same letters as figure 8(b), the similarity between figures 8(a) and 8(b) is far more apparent than that between figures 6(a) and 6(b). Therefore, providing fixed positions for certain characters allows the dynamic set of characters to appear to retain some form of stability even though it remains entirely dynamic. The increased appearance of stability enables the user to develop a

30   familiarity with the placement of the eight fixed characters, which, in turn, will improve the text entry speed.

In general, the positioning of characters in the dynamic region is accomplished by placing each successive character in L as close to the cursor position as possible, given the

remaining unfilled slots. In the event that several positions in the dynamic part are at identical distances from the cursor, the order of filling these positions is at the discretion of the programmer.

The basic algorithm for positioning characters in a fluctuating layout is as follows:

5   Let γ = the last character entered

Let L = the alphabetic sequence sorted by order of likelihood to follow to γ

Let n = length(L)

repeat i = n times

place L(i) as close to the cursor as possible

10   end repeat

The algorithm does not indicate what to do when there are multiple available positions at equal distances from the cursor. This is, in fact, a design choice. To explore the issue, let us consider a three-row, nine-column layout where the cursor is in the top left corner at the time the layout is rearranged. Figure 9 shows a variety of possible ways to

15   position characters according to the algorithm, using the alphabetic sequence corresponding to the space character. The layouts on the right are the result of positioning characters in the sequence atop the figure, according to the patterns on the left.

In this specific case, positions in the top row are superior to other positions because they only require the use of one arrow key to reach, whereas positions in rows 2 and 3 require

20   the use of two arrow keys. For example, in the last layout in Figure 9f, moving the cursor to the letter H in row 1 only requires the right arrow key, while moving to the W in row 2 requires the right and down arrow keys. In other words, positions in row 1 and column 1 are accessible via a less complex route than positions in rows 2 and 3, starting from column 2. It might make sense to fill row 1, and column 1, positions before others at identical distances

25   form the cursor. As a modification to the algorithm, this could be generalized as follows: Given more than one available position at identical distances from the cursor, positions in the same row or column as the cursor should be filled before other positions.

Another distinction between identically distanced positions is the number of possible paths from the cursor to the position. As a modification to the algorithm, this could be

30   generalized as follows: Given more than one available position at identical distances from the cursor, fill positions in increasing order of the number of paths from the cursor to the position.

Another possibility is that the choice between several identically distanced positions has no significant effect on text entry speed.

Another embodiment of the invention allows the user to be able to switch languages. The language switch may result in a different fixed layout 12 as well as a different table of

5   next character probabilities. However, the basic method for updating the dynamic region as used for a language other than English remains the same.

In yet another embodiment, the dynamic region is not generated by a single preceding character, but by a preceding string of characters. The use of this string prediction method makes the prediction more accurate than a simple single-character based prediction. An

10   example of this is illustrated in table 1, below.

(a)   _ _ $\underline{C}$ _   TACT?   (b) $\underline{O}$ $\underline{N}$ $\underline{C}$ _   ONCE!
                           SUCH?
                           RICE?

In column (a) there are several candidates for the fourth position, while in column
(b) there is only one.

15   The dynamic region 14 may have a different layout to the 3x3 matrix above in order to adapt to specific applications. Some alternate possible layouts are illustrated in figures 10(a), 10(b), and 10(c) above. The home position in each of the layouts is slightly different, yet no dynamic key is more than two moves of the cursor away from it. To increase the number of dynamic keys or the distance that they are from the home key reduces the

20   effectiveness of using a hybrid keyboard by increasing the number of options available to the user, which increases the decision time. The excess keys reduce the effectiveness of the hybrid keyboard, but does not render it useless.

The location of the cursor position within the dynamic part is a design option. There are tradeoffs associated with different choices. One consideration is that the home

25   position be no greater than a certain distance from any of the dynamic positions. Another consideration is that the home position be adjacent to an edge of the layout for the facilitation of implicit selection of the space character (discussed below).

The positioning of this virtual space bar with respect to the layout is a design option contingent to a degree on the choice of cursor home position. Another option is whether or not to actually display the virtual space bar.

The position of the dynamic layout in the overall layout may also be changed as
5   shown by the shaded squares of Figure 11. Furthermore, the dynamic keys may be interspersed throughout part or all of the layout.

Flow charts for implementing an embodiment of the invention are shown in figures 13 through 15.

In the previous embodiments, an underlying assumption of is that the number of
10 characters in the language in question's alphabet is small, that is, in the order of the size of the English alphabet. However, for many Asian languages, for example Chinese and Japanese, the number of characters is much larger that the English language.

Therefore, in an alternate embodiment for such languages, if the various strokes used to construct characters are used as primitive elements in previous embodiments, as opposed
15 to the characters themselves, the problem of having a large number of characters is diminished.

The number of unique strokes in Chinese and Japanese, and other Asian languages with very large character sets is in the order of the size of the English alphabet (typically in the order of fifty). In order to utilize the languages such as Chinese or Japanese there are
20 two additional requirements.

First, data on stroke, rather than character, sequence probabilities are used to control the display of strokes in the dynamic region. This is greatly simplified by the fact that in these languages there is often a very strictly observed stroke order in the construction of characters.

25 Second, a stroke assembly software mechanism is used. The software mechanism includes character prediction, whereby as the user enters strokes the most likely candidate characters are suggested.

Although the invention has been described with reference to certain specific
30 embodiments, various modifications thereof will be apparent to those skilled in the art without departing from the spirit and scope of the invention as outlined in the claims appended hereto.

THE EMBODIMENTS OF THE INVENTION IN WHICH AN EXCLUSIVE
PROPERTY OR PRIVILEGE IS CLAIMED ARE DEFINED AS FOLLOWS:

1.   A method for inputting characters in a device having a character display, and a
navigation key responsive to user manipulation for navigating a cursor in the character
display, said method comprising the steps of:

     (d) displaying a character set having a predetermined layout on said display, said
layout being fixed for one or more character input;

     (e) displaying a subset of said character set in a dynamic layout wherein said subset
of characters are determined by at least one preceding input character; and

     (f) selecting at least one of said characters from said layouts by navigating said
cursor to a desired character.

2.   A character-input device having a character display and navigation key responsive to
a user manipulation for navigating a cursor in said character display, comprising:

     (a) a character set having a predetermined layout for display on said display, said
layout being fixed for one or more character input;

     (b) a subset of said character set for display in a dynamic layout on said display
wherein said subset of characters are determined by at least one preceding input
character; and

     (c) a selector for selecting at least one of said characters from said layouts by
navigating said cursor to a desired character.

SUBSTITUTE SHEET (RULE 26)

Figure 1

Figure 2

14

17

15
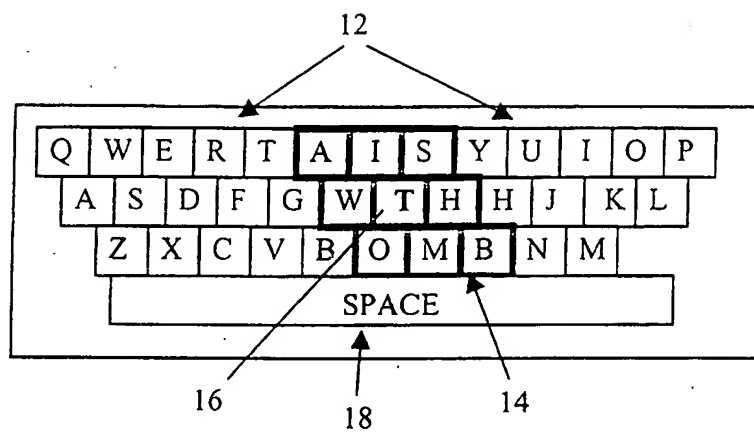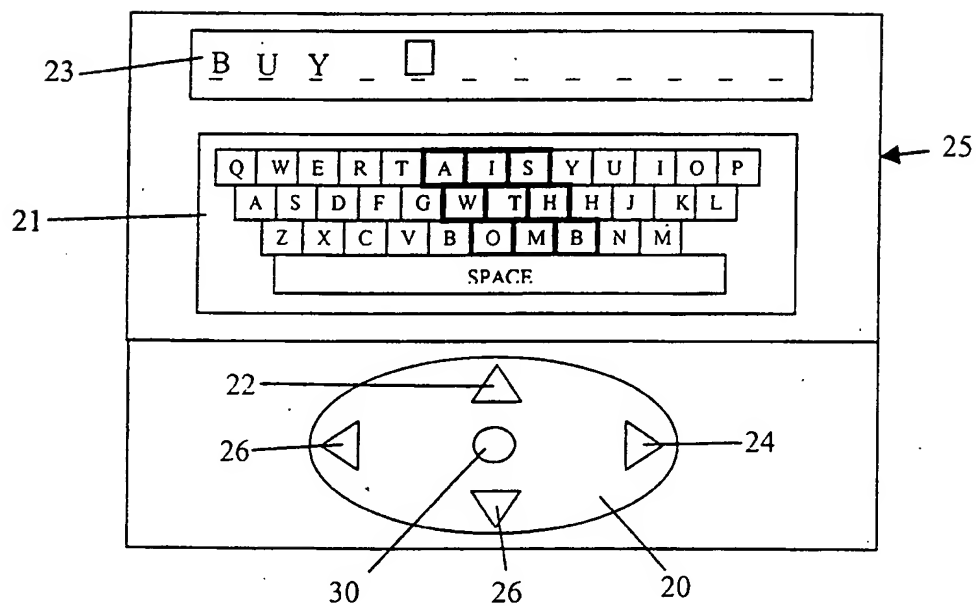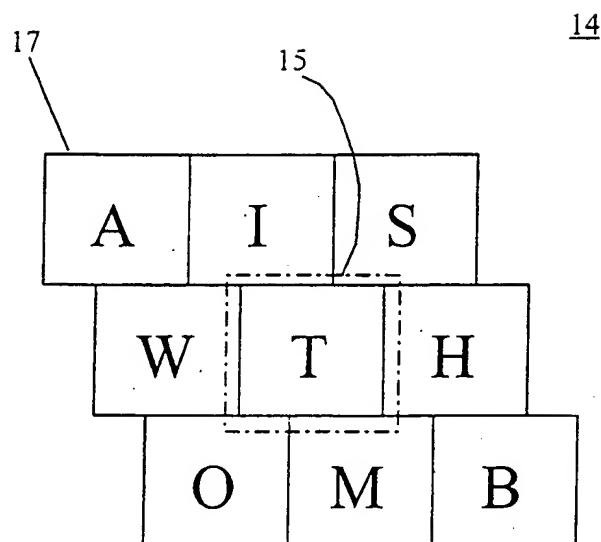
Figure 3

A   NRTSLDIYCVMKBGPUFW_XZHJAEOQ

B   EUOALRI_JSBYTMVWCDFGHKNPQXZ

C   HAOEKTLRI_UCYSNQBDFGJMPVWXZ

D   _EIOASRUYDLGVFTNWBJMCHKPQXZ

E   _RNDASELTYMCVIWFPXGOKHQBUZJ

F   OIREA_TFLUSYMBCDGHJKNPQVWXZ

G   _EHORIAULSNGTYBMWCDFJKPQVXZ

H   EAI_OTRUYSWLPHMBCDFGJKNQVXZ

I   NTSLCRDMGEVOKFPAZXBU_QIHJWY

J   UOEIABCDFGHJKLMNPQRSTVWXYZ_

K   E_INSLYOUAMPBCDFGHJKQRTVWXZ

L   _ELIOADYSUTFKPMWVRGCBNHJQXZ

M   E_AOIUPSBMYFLNTCDGHJKQRVWXZ

N   D_GETOSIYACLKNUJFHQVBRWXMPZ

O   URNMTWO_LSPVKDIFBACYEGZJXHQ

P   ELOA_RIUPSTHYWCFBDGJKMNQVXZ

Q   UABCDEFGHIJKLMNOPQRSTVWXYZ_

R   _EOSIATYDNULRMGKCVHFBPWJQXZ

S   _TEHIOAUSPKLCYWMNBQGDFJRVXZ

T   H_EOAISRTUYWLCBGNDFMZJKPQVX

U   TRSN_LGCMEIBPDAFYOZQXHJKUVW

V   EIOAYUBCDFGHJKLMNPQRSTVWXZ_

W   AHI_EONRSLDTBUKCFGJMPQVWXYZ

X   TP_AICEUOBDFGHJKLMNQRSVWXYZ

Y   _OESIPABLTWGHMUCDFJKNQRVXYZ

Z   EZLASYIO_BCDFGHJKMNPQRTUVWX

_   TAWSHFBMCOLPDRNGYEIKUVJQZX_

Figure 4

	A	B	C	D	E	F	G	H	...									S	T	U	V	W	X	Y	Z
A																									
B																									
C																									
D																									
E																									
F																									
G																									
H																									
.																									
.																									
S																									
T																									
U																									
V																									
W																									
X																									
Y																									
Z																									
AA																									
AB																									
.																									
.																									
.																									
AZ																									
BA																									
BB																									
.																									
.																									
.																									
ZZ																									

Figure 5a

Figure 5b

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	SPACE	TOTAL
A	2	144	308	382	1	67	138	9	322	7	146	664	177	1576	1	100	0	802	683	785	87	233	57	14	319	12	50	7086
B	136	14	0	0	415	0	0	0	78	18	0	98	1	0	240	0	0	88	15	7	256	1	1	0	13	0	36	1417
C	368	0	13	0	285	0	0	412	67	0	178	108	0	1	298	0	1	71	7	154	34	0	0	0	9	0	47	2053
D	106	1	0	37	375	3	19	0	148	1	0	22	1	2	137	0	0	83	95	3	52	5	2	0	51	0	2627	3770
E	670	8	181	767	470	103	46	15	127	1	35	332	187	799	44	90	9	1314	630	316	8	172	106	87	189	2	4904	11612
F	145	0	0	0	154	86	0	0	205	0	0	69	3		429	0	0	188	4	102	62	0	0	0	4	0	110	1561
G	94	1	0	0	289	0	19	288	96	0	0	55	1	31	135	0		98	42	6	57	0	1	0	2	0	686	1901
H	1164	0	0	0	3155	0	0	1	824	0	0	5	1		487	2	2	91	8	165	75	165	8	0	32	18	715	6733
I	23	7	304	260	189	56	233	0	1	0	86	324	255	1110	88	42		272	484	558	5	0	0	15	0	0	4	4501
J	2	0	0	0	31	0	0	0	9	0	0	0	0	0	41	0	0	0	0	0	56	0	0	0	0	0	0	139
K	2	0	0	0	337	0	0	0	127	0	0	10	1	82	3	1	0	0	0	0	3	0	0	0	8	0	309	933
L	332	4	6	289	591	59	7	0	390	0	38	546	30	1	344	34	0	11	50	74	81	17	19	0	276	0	630	3900
M	394	53	0	0	530	6	0	0	165	0	0	4	28	4	289	77	0	0	2	2	85	0	0	0	19	0	454	2160
N	100	2	98	1213	512	5	771	5	135	8	63	80	0	54	349	0	3	2	148	378	49	3	2	2	115	5	1152	5249
O	65	67	61	119	34	80	9	1	88	3	123	218	417	598	336	138	0	812	195	415	1115	136	398	2	47	0	294	5776
P	142	0	1	0	280	1	0	24	97	0	0	169	0	0	149	64	0	110	48	40	68	0	3	0	14	0	127	1337
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	66	0	0	0	0	0	0	66
R	289	10	22	133	1139	13	59	21	309	0	53	71	65	106	504	9	8	69	318	190	89	22	5	0	145	0	1483	5124
S	196	9	47	0	626	0	1	328	214	0	57	48	31	16	213	107	0	0	168	754	175	0	32	0	34	0	2228	5292
T	259	2	31	1	583	1	2	3774	252	0	0	75	1	2	331	0	0	187	209	154	132	0	84	0	121	1	2343	8545
U	45	53	114	48	71	10	148	0	65	0	0	247	87	278	3	49	1	402	299	492	0	0	0	1	7	3	255	2678
V	27	0	0	0	683	0	0	0	109	0	0	0	0	0	33	0	0	0	0	0	1	0	0	0	11	0	0	864
W	595	3	9	6	285	0	0	472	10	0	1	12	0	103	264	0	0	35	21	4	2	0	0	0	0	0	326	2503
X	17	0	9	0	9	0	0	0	32	0	0	7	1	0	1	22	0	0	81	23	8	0	0	0	0	0	21	120
Y	11	10	0	0	152	0	1	1	32	0	0	7	1	0	339	16	0	0	81	2	1	0	2	0	0	0	1171	1827
Z	3	0	0	0	26	0	0	0	2	0	0	4	0	0	2	0	0	0	3	0	0	0	0	0	3	9	2	54
SPACE	1882	1033	864	515	423	1059	453	1388	237	93	152	717	876	478	721	588	42	494	1596	3912	134	116	1787	0	436	2	0	19974
TOTAL	7069	1418	2059	3770	11645	1549	1906	6739	4483	131	932	3885	2163	5241	5781	1339	66	5129	5278	8536	2701	870	2507	121	1855	52	19974	107199

I	R	L
E	H	O
S	A	U

Figure 6a

S	H	A
E	T	I
P	O	U

Figure 6b

A	E	I
O		U
S	N	L

Figure 7
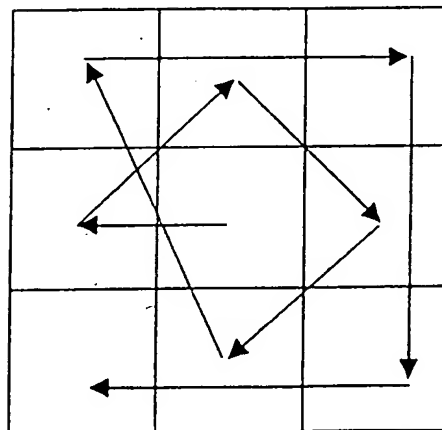
Figure 12

A	E	I
O	H	U
S	R	L

Figure 8a

A	E	I
O	T	U
S	H	P

Figure 8b

SPACE: TAISWHOMBCFDPLNREGYUVKJQZX
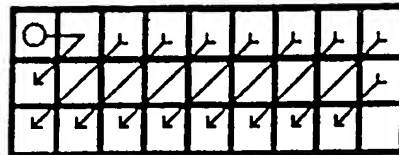
Figure 9a

🔟	A	S	O	C	P	R	Y	K
I	W	M	F	L	E	U	J	Z
H	B	D	N	G	V	Q	X	

Figure 9b
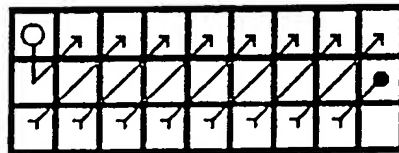
Figure 9c

🔟	I	H	B	D	N	G	V	Q
A	W	M	F	L	E	U	J	X
S	O	C	P	R	Y	K	Z	

Figure 9d
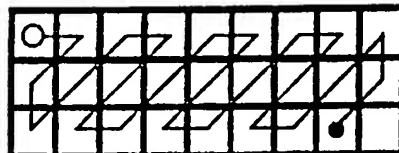
Figure 9e

🔟	A	H	O	D	P	G	Y	Q
I	W	M	F	L	E	U	J	Z
S	B	C	N	R	V	K	X	

Figure 9f
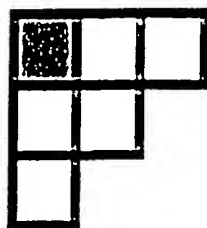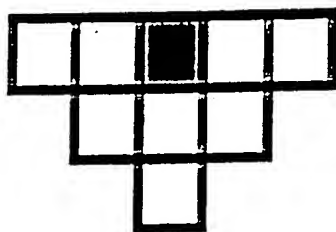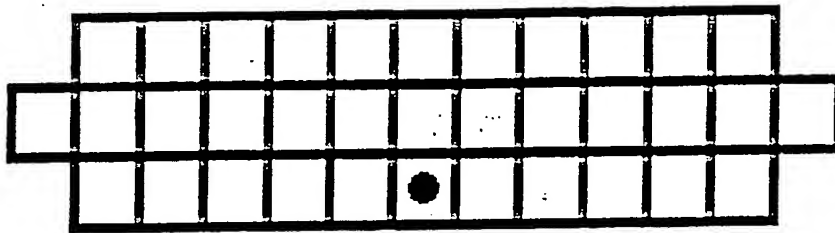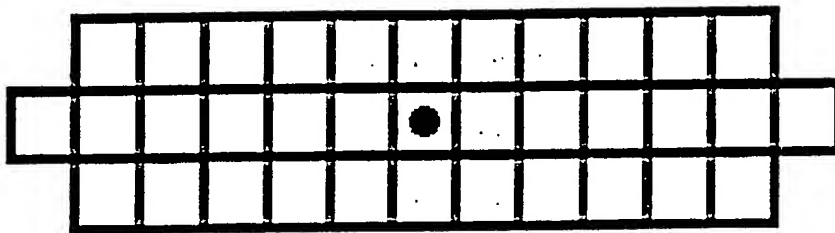
Figure 10a   Figure 10b   Figure 10c
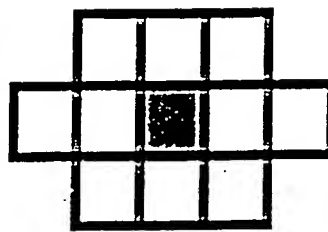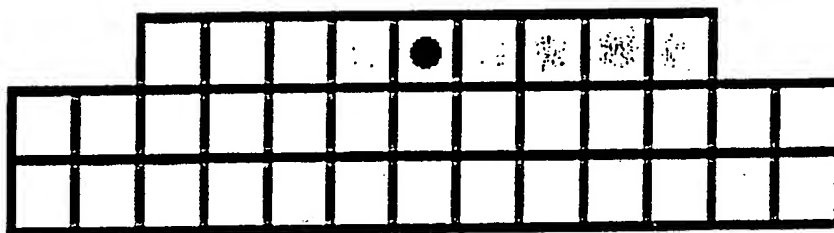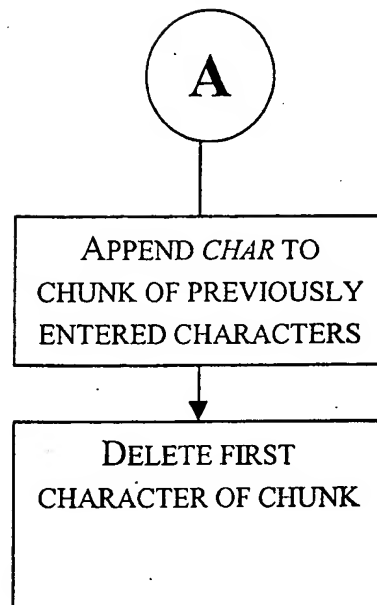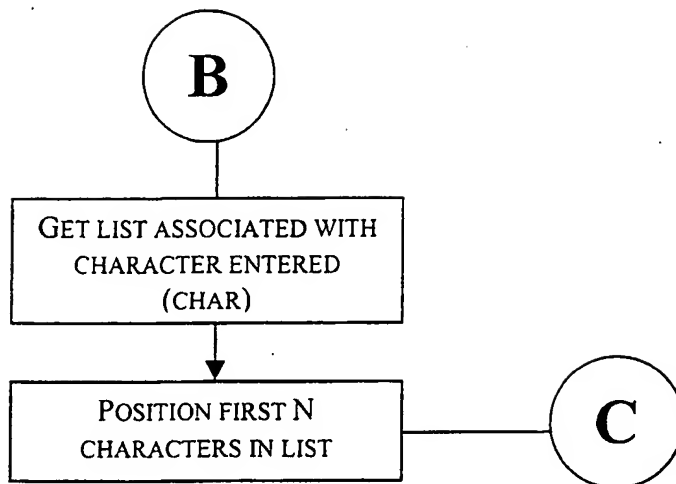
Figure 11a

Figure 11b

Figure 11c

Figure 11d

Figure 11e

DISPLAY INITIAL
CHARACTER SET IN
DYNAMIC REGION

DISPLAY CURSOR AT
HOME POSITION

SET *CHAR* TO CHARACTER AT
HOME POSITION

USER MOVED
CURSOR

WAIT FOR
INPUT

USER
PERFORMED A
SELECTION

SET *CHAR* TO
CHARACTER AT
CURSOR POSITION

PERFORM CONTROL
FUNCTION

YES

IS *CHAR* A
CONTROL
FUNCTION?

DISPLAY CURSOR
AT NEW POSITION

NO

IS CURSOR ON
SPACE CHARACTER
POSITION?

YES

SET *CHAR* TO
SPACE
CHARACTER

DISPLAY *CHAR* IN
DISPLAY AREA

NO

IS CURSOR AT
A CONTROL
FUNCTION

NO

A

UPDATE CHUNK OF
PREVIOUSLY ENTERED
CHARACTERS

SET *CHAR* TO
CONTROL FUNCTION
AT CURSOR POSITION

YES

UPDATE LAYOUT
OF DYNAMIC
REGION

IS *IMPLICIT
SELECTION OF
CONTROL
FUNCTIONS* TRUE?

PERFORM CONTROL
FUNCTION

B

Figure 13(a)

Figure 13(b)

B

GET LIST ASSOCIATED WITH
CHARACTER ENTERED
(CHAR)

POSITION FIRST N
CHARACTERS IN LIST

C

Figure 14a

B

LOCATE ROW OF MATRIX WITH HEADING
THAT MATCHES CHUNK OF PREVIOUSLY
ENTERED CHARACTERS

SORT ROW OF CELLS AND CORRESPONDING
COLUMN HEADINGS IN NON-INCREASING ORDER
OF CELL VALUES

SET LIST TO SEQUENCE OF
SORTED COLUMN HEADINGS

POSITION FIRST N
CHARACTERS IN LIST

C

Figure 14b

Figure 15